

Prueba - Conversor de monedas

En esta prueba validaremos nuestros conocimientos del método `fetch`.

Lee todo el documento antes de comenzar el desarrollo **individual**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Descripción

Aplicando los conceptos y herramientas aprendidas hasta ahora, en esta prueba deberás programar un conversor de monedas a partir de un monto en pesos chilenos. Para esto será necesario consultar la API mindicador.cl usando el método `fetch`.



En caso de que la API no se encuentre disponible, puedes utilizar la versión offline que se encuentra en la plataforma, en el archivo con nombre "Apoyo prueba - Conversor de monedas".

A continuación te mostramos lo que debes maquetar:

Conversor de monedas Nacional

Pesos CLP

Ingrese el monto en CLP

Moneda a convertir:

Seleccione moneda ▾

Buscar

...

Imagen 1. Maqueta general de la prueba
Fuente: Desafío Latam

En donde tenemos:

- Un *input* para ingresar la cantidad de pesos chilenos a convertir
- Un *select* para elegir la moneda a convertir
- Un botón para iniciar el proceso de consulta y renderización de datos

The screenshot shows a web interface titled "Conversor de monedas Nacional". It features a central grey box with the following elements: a label "Pesos CLP" above an input field containing "25000"; a label "Moneda a convertir:" above a dropdown menu showing "Dolar"; a blue button labeled "Buscar"; and a result label "Resultado: \$31.27".

Imagen 2. Búsqueda realizada con éxito
Fuente: Desafío Latam

Luego, utiliza una librería de JavaScript de gráficas para mostrar un historial de los últimos 10 días del valor de la moneda a convertir seleccionada.



Imagen 3. Conversión con historial
Fuente: Desafío Latam

Requerimientos

1. Se obtienen los tipos de cambio desde mindicador.cl (**1 punto**).
2. Se calcula correctamente el cambio y se muestra en el DOM (**3 puntos**).
3. El select implementa más de un tipo de moneda (con 2 es suficiente), todos los cambios funcionan correctamente (**3 puntos**).
4. Se usa *try catch* para ejecutar el método fetch y capturar los posibles errores mostrando el error en el DOM en caso de que haya problemas (**2 puntos**).

Tip: Recuerda que la guía de APIs hay un capítulo asociado a “Sentencias Try y Catch”

5. Se Implementa el gráfico pedido (**1 punto**).

Tip: En la guía de APIs hay un capítulo asociado a “Plugins de JavaScript (Chart Js)”. Utiliza esto como referencia para generar el gráfico.



¡Mucho éxito!